



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/814,324	03/21/2001	Motohiro Kawahito	JP919990309US1	3779

7590 04/06/2005
Casey August
Intellectual Property Law Dept.
IBM Corporation
P.O. Box 218
Yorktown Heights, NY 10598

EXAMINER

NAHAR, QAMRUN

ART UNIT PAPER NUMBER

2191

DATE MAILED: 04/06/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/814,324	Applicant(s) KAWAHITO ET AL.	
	Examiner Qamrun Nahar	Art Unit 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 11 February 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-18 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-8, 10 and 12-18 is/are rejected.
- 7) ☒ Claim(s) 9 and 11 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is in response to the RCE filed on 2/11/05.
2. The objections to claim 18 are withdrawn in view of applicant's amendment.
3. The rejection under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention to claims 3, 5 and 17-18 is withdrawn in view of applicant's amendment and remarks/arguments.
4. Claims 1, 6-8, 10, 12-13 and 15-18 have been amended.
5. Claims 1-18 are pending.
6. Claims 1-2, 4, 6-7, 10 and 12-16 are rejected under 35 U.S.C. 102(e) as being anticipated by Dunn (U.S. 6,247,172).
7. Claims 3, 5, 8, and 17-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dunn (U.S. 6,247,172) in view of Ghosh (U.S. 6,412,109).
8. Claims 9 and 11 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Amendment

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

Art Unit: 2191

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. Claims 1-2, 4, 6-7, 10 and 12-16 are rejected under 35 U.S.C. 102(e) as being anticipated by Dunn (U.S. 6,247,172).

Per Claim 1 (Amended):

The Dunn patent discloses:

- **a compiler for converting source code for a program written in a programming language into object code in a machine language for a target machine** ("FIG. 7 is a simplified block diagram of a compiler embodiment of the translation system according to the present invention. This is another embodiment of the translation system described in FIG. 4. A compiler can be thought of as a translator, translating a source program into compiled code. Compilers can aggressively optimize to produce the most efficient code possible." in column 7, lines 20-30; and see Figure 7; This *embodiment* teaches a compiler, where the source code is translated into object code in a machine language for a *target machine*)

- **an optimization execution unit for performing an optimization process for an object program written in a machine language for a target machine** ("The compilation system 70 includes a compiler 74 that converts a source program 72 written in a high order language, such as C, into compiled code 78 operable on the target platform comprising the target hardware and the target operating system 84. The compiler 74 creates compiled code 78, including the optimized

Art Unit: 2191

compiled code 80 as well as recovery blocks 82. The compiler produces a set of recovery blocks 82, up to one for each potentially trapping instruction in the optimized compiled code 80.” in column 7, lines 31-39)

- and a program modification unit for modifying said object program in order to absorb a difference in content between the point of origin of an exception process, which occurs in response to the execution on the target machine of a command in said object program, and an exception handler whereat said exception process is performed (“When the optimized compiled code 80 *executes on the target operating system 84*, a synchronous exception may result. If a synchronous exception occurs, the target operating system 84 invokes a runtime recovery procedure 86, linked to the application as part of the runtime library. The runtime recovery procedure then *restores the target machine state* using the recovery blocks 82.” in column 7, lines 40-46 (emphasis added); the recovery block compensates for the difference in content from the point where the exception occurs and the exception handler).

Per Claim 2:

The Dunn patent discloses:

- wherein, if there is a difference in content between the point of origin of an exception process, which occurs in response to the execution of a command in said object program, and a location whereat said exception process is performed, said program modification unit

Art Unit: 2191

generates compensation code to compensate for said difference, and inserts said compensation code into said object program (column 7, lines 35-46).

Per Claim 4:

The Dunn patent discloses:

- wherein, before said optimization execution unit performs said optimization process in said object program, said program modification unit divides said command that may cause an exception process into a command portion for determining whether an exception process has occurred, and a command portion for actually causing an exception process; and wherein, when an exception process occurs, said program modification unit modifies said object program to shift program control to said command portion that actually caused said exception process (column 6, lines 9-32).

Per Claim 6 (Amended):

This is a system version of the claimed compiler discussed above, claim 1, wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above.

Thus, accordingly, this claim is also anticipated by Dunn.

Per Claim 7 (Amended):

This is a method version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth

Art Unit: 2191

above, including “moving program control to a portion whereat said exception process is performed” (column 6, lines 10-15). Thus, accordingly, this claim is also anticipated by Dunn.

Per Claim 10 (Amended):

The Dunn patent discloses:

- **an optimization method for optimizing a program to increase processing efficiency**

(column 7, lines 20-30)

- **dividing software code, in an object program, that may cause an exception process into software code for determining whether an exception process has occurred and software code for actually causing an exception process** (column 7, lines 31-39; and column 6, lines 36-67 to column 7, lines 1-6)

- **specifying said code obtained at said division step as branches of a control flow graph; designing said control flow graph so that when an exception process occurs, program control is shifted to said code that actually caused said exception process; and performing said optimization process for said object program that has been modified** (column 7, lines 40-46; column 6, lines 36-67 to column 7, lines 1-6; and see Figure 6).

Per Claim 12 (Amended):

Art Unit: 2191

This is a computer program version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also anticipated by Dunn.

Per Claim 13 (Amended):

This is a computer executable program version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also anticipated by Dunn.

Per Claim 14:

The Dunn patent discloses:

- wherein said function for determining whether an exception process has occurred is provided by a condition branch (column 6, lines 56-67 to column 7, lines 1-6).

Per Claim 15 (Amended):

This is a storage medium version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also anticipated by Dunn.

Per Claim 16 (Amended):

This is a apparatus version of the claimed compiler discussed above (claims 1 and 2), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also anticipated by Dunn.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. Claims 3, 5, 8, and 17-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dunn (U.S. 6,247,172) in view of Ghosh (U.S. 6,412,109).

Per Claim 3:

The rejection of claim 1 is incorporated, and Dunn further teaches wherein said program modification unit includes: a pre-processor for, before said optimization execution unit performs said optimization process, examine a command that may cause an exception process in said object program to determine whether an exception process has occurred, and performing an inherent process when it is found an exception process has occurred (column 7, lines 31-39), and a post-processor for examining, in said object program that has been optimized by said optimization execution unit, said command that may cause an exception process to determine whether a difference in content exists between said command that may cause said exception process and a location whereat said exception process is performed, and for, when a difference

Art Unit: 2191

exists, generating a compensation code, to be used to compensate for said difference, and a code for, after said compensation code is obtained, moving program control to an exception handler whereat said exception process is performed (column 7, lines 40-46 and column 6, lines 19-35). Dunn does not explicitly teach try-catch blocks for handling exceptions. Ghosh teaches try-catch blocks for handling exceptions (column 5, lines 26-34).

It would have been obvious to one having ordinary skill in the computer art at the time of the invention was made to modify the compiler disclosed by Dunn to include try-catch blocks for handling exceptions using the teaching of Ghosh. The modification would be obvious because one of ordinary skill in the art would be motivated to increase the amount of code that can be optimized, even code that is associated with a try-catch block (Ghosh, column 6, lines 51-60).

Per Claim 5:

This is another version of the claimed compiler discussed above (claims 3 and 4), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claim 8 (Amended):

This is a method version of the claimed compiler discussed above (claims 1 and 3), wherein all claim limitations also have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, this claim is also obvious.

Per Claim 17 (Amended):

Art Unit: 2191

The rejection of claim 3 is incorporated, and Dunn further teaches wherein the post-processor registers an address of the point at which the exception process occurs (column 6, lines 19-35).

Per Claim 18 (Amended):

The rejection of claim 17 is incorporated, and Dunn further teaches wherein the post-processor registers an address of the exception handler (column 6, lines 19-35).

Allowable Subject Matter

13. Claims 9 and 11 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

14. The following is a statement of reasons for the indication of allowable subject matter:

The cited prior art taken alone or in combination fail to teach, in combination with the other claimed limitations, wherein said step of determining whether a difference in content exists between said command that may cause an exception process and said exception handler whereat said exception process is performed includes a step of: removing said basic block prepared for said command that may cause an exception process when no difference in content exists as recited in claim 9; and determining whether code for compensating for a difference in content between the point of origin of an exception process and code for actually causing said exception process have been generated in a block that includes code for the actual performance of said

Art Unit: 2191

exception process after the optimization process has been run; and using said control flow graphs, when said code for compensating for said content difference is not generated, for synthesizing said two code sets to obtain said code arrangement that existed before said code was divided as recited in claim 11.

The closest cited prior arts, the combination of Dunn (U.S. 6,247,172) and Ghosh (U.S. 6,412,109) teaches a method of optimizing code containing exceptions. However, the combination of Dunn (U.S. 6,247,172) and Ghosh (U.S. 6,412,109) fails to teach wherein said step of determining whether a difference in content exists between said command that may cause an exception process and said exception handler whereat said exception process is performed includes a step of: removing said basic block prepared for said command that may cause an exception process when no difference in content exists as recited in claim 9; and determining whether code for compensating for a difference in content between the point of origin of an exception process and code for actually causing said exception process have been generated in a block that includes code for the actual performance of said exception process after the optimization process has been run; and using said control flow graphs, when said code for compensating for said content difference is not generated, for synthesizing said two code sets to obtain said code arrangement that existed before said code was divided as recited in claim 11.

Response to Arguments

15. Applicant's arguments filed on 12/27/04 have been fully considered but they are not persuasive.

In the remarks, the applicant argues that:

Art Unit: 2191

a) With regard to the rejection of Claims 1-18 over the prior art, these claims patentably distinguish over the prior art because the references do not disclose or suggest the exception handling process as described in independent Claims 1, 6, 7, 8, 10, 12, 13, 15 and 16. In order to best understand this, it may be helpful if Applicants briefly review the present invention and the prior art.

The present invention generally relates to a procedure for transforming program code written for one machine into program code suitable for a second machine. More particularly, this invention provides effective optimization for such a program where that program may cause or include an exception process.

One challenge involved in emulation of code designed for one architecture (often a legacy platform) on a machine of a different architecture (referred to as the target machine or target platform) occurs in systems that support the handling of exceptions. When the conditions that would cause an exception in the original code occur, the translation system may need to ensure that the modeled state (the target machine state is consistent with the original, or legacy, machine state. This consistency is difficult to maintain in an aggressively optimizing translator. This is because optimization can re-arrange the order of execution of code, and when an exception occurs, the target machine state may not match the state expected by the old program.

The present invention effectively addresses this challenge by providing a program modification unit for modifying the object program in order to absorb a difference in content between the point of origin of an exception process, which occurs in response to the execution on the target machine of a command in the object program, and an exception handler at where the exception process is performed.

Dunn does not disclose or teach the program modification unit of this invention, and, more specifically, Dunn does not disclose or suggest compensating for the difference in the content of the code between the start of the exception process on the target machine and the exception handler.

In the Office Action, the Examiner argued that the recovery block 42 of Dunn corresponds to the program modification unit of the present invention. There are, though, important differences between the recovery block 42 of Dunn and the program modification unit of this invention.

More specifically, in this invention, the use of the program modification unit "to absorb a difference..." has a different objective from Dunn's recovery block 42. The machine state of the compensation code of this invention is the "target machine" state at an exception handler. However, the machine state of the recovery block of Dunn is the "legacy machine" state at the code that is throwing the current exception. Therefore, the program modification unit of this invention and the recovery block 42 of Dunn are at different locations and in different states. With the procedure described in Dunn, the recovery block contains instructions that complete all functions necessary to restore the target machine state to "the legacy machine state" (Dunn, column 3, line 13).

The legacy machine state is the machine state that could have resulted had the target platform executed "target code not optimized by the compiler" (Dunn, column 3, line 52). In contrast, with the procedure of this invention, program control is shifted, preferably through compensation codes, to the exception handler in order to "absorb the difference between the point of origin of the execution occurrence point on the target machine and the exception

Art Unit: 2191

handler." The present invention, for instance, may perform the algorithms shown in Figures 4 to 7 in order to compensate a register image between the point of origin of the exception occurrence points on the target machine and the exception handler. With particular reference to Figure 10 of this application, Dunn's approach does not generate the "copy i and j variable values to R1 and R2" in the recovery block.

Independent Claims 1, 6, 7, 8, 10, 12, 13, 15 and 16 describe the above-discussed feature of this invention relating to the operation of the program modification unit. Claims 1 and 6 describe a program modification unit for modifying the object program in order to absorb a difference in content between the point of origin of an exception process, which occurs in response to the execution on the target machine of a command in said object program, and an exception handler whereat said exception process is performed.

Claim 7 describes the step of preparing a basic block that includes a portion for ...

...

Claims 12, 15 and 16 describe a process for, when a difference in content exists between the point of origin of an exception process on the target machine and an exception handler whereat said exception process is performed, generating in a basic block compensation code for compensating for said difference. Analogously, Claim 13 describes a function for, when an exception process has occurred on a target machine, shifting program control to an exception handler whereat said exception process is run, and for when a difference in content exists between the point of origin of said exception process on the target machine and said exception handler whereat said exception process is run, compensating for said difference before program control is shifted to said exception handler.

Examiner's response:

a) Examiner strongly disagrees with applicant's assertion that Dunn fails to disclose the claimed limitations recited in claims 1, 6, 7, 10, 12, 13, 15 and 16. Dunn clearly shows each and every limitation in claims 1, 6, 7, 10, 12, 13, 15 and 16.

Dunn teaches a program modification unit for modifying said object program in order to absorb a difference in content between the point of origin of an exception process, which occurs in response to the execution on the target machine of a command in said object program, and an exception handler whereat said exception process is performed ("When the optimized compiled code 80 *executes on the target operating system 84*, a synchronous exception may result. If a synchronous exception occurs, the target operating system 84 invokes a runtime recovery procedure 86, linked to the application as part of the runtime library. The runtime recovery procedure then *restores the target machine state* using the recovery blocks 82." in column 7, lines 40-46 (emphasis added) and see Figure 7; the recovery block compensates for the difference in content from the point where the exception occurs and the exception handler).

In addition, see the rejection above in paragraph 10 for rejection to claims 1, 6, 7, 10, 12, 13, 15 and 16.

In the remarks, the applicant argues that:

b) The other references of record have been reviewed. These other references, even when considered in combination, fail to disclose or suggest the above-discussed exception handling process. In particular, Ghosh was cited for its disclosure of try-catch block for handling

Art Unit: 2191

exceptions. This reference, though, does not teach the use of an exception handler to identify a point in a process on the target machine with respect to which differences should be measured or compensated for, as described in Claims 1, 6, 7, 8, 10, 12, 13, 15 and 16.

Because of the above-discussed differences between Claims 1, 6, 7, 8, 10, 12, 13, 15 and 16 and the prior art, and because of the advantages associated with those differences, Claims 1, 6, 7, 8, 10, 12, 13, 15 and 16 patentably distinguish over the prior art and are allowable. Claims 2-5, 17 and 18 are dependent from Claim 1 and are allowable therewith. Similarly, Claim 9 is dependent from, and is allowable with, Claim 8; and Claims 11 and 14 are dependent from and are allowable with Claims 10 and 13 respectively.

Examiner's response:

b) In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Ghosh is relied upon for try-catch blocks for handling exceptions, not for "the use of an exception handler to identify a point in a process on the target machine with respect to which differences should be measured or compensated for".

Furthermore, claims 9 and 11 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims, see paragraph 13 above.

In addition, see the rejection above in paragraph 10 for rejection to claims 1-2, 4, 6-7, 10 and 12-16; and paragraph 12 for rejection to claims 3, 5, 8, and 17-18.

Conclusion

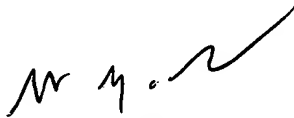
16. Any inquiry concerning this communication from the examiner should be directed to Qamrun Nahar whose telephone number is (571) 272-3730. The examiner can normally be reached on Mondays through Fridays from 9:30 AM to 6:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam, can be reached on (571) 272-3695. The fax phone number for the organization where this application or processing is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QN
March 25, 2005


WEI Y. ZHEN
PRIMARY EXAMINER